

# Re: Async Socket IO Question

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework/2006-10/msg00020.html>

---

- *From:* "Chris Mullins" <[cmullins@xxxxxxxxxx](mailto:cmullins@xxxxxxxxxx)>
  - *Date:* Wed, 4 Oct 2006 11:31:28 -0700
- 

I think you're mostly bumping into Socket Timeout issues.

If you kill the process of your client app, the TCP session isn't cleanly shut down. Your server still thinks the connection exists. You can verify this using "NetStat -a". You'll see your connection still in there.

When your server sends to the client app, that send happens just fine (the TCP Session still there). After a few moments, your TCP send will timeout, and you'll get an error. At this point the TCP Session is torn down.

Just because it's amusing, I have had more bugs in socket shutdown code than all the other areas of networking put together. There are so many ways, and so many conditions, that can cause a TCP session to be torn down that it's just depressing.

Here's an MS KB article that goes into how to adjust your timeouts:  
<http://support.microsoft.com/?kbid=170359>

Note: I don't recommend adjusting your timeouts – but you do need to have a solid understanding of what's going on.

--

Chris Mullins, MCSD.NET, MCPD:Enterprise  
<http://www.coversant.net/blogs/cmullins>

"EmeraldShield" <[emeraldshield@xxxxxxxxxxxxxxxxxxxx](mailto:emeraldshield@xxxxxxxxxxxxxxxxxxxx)> wrote in message  
[news:%23Q\\$abF%235GHA.4644@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:%23Q$abF%235GHA.4644@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)

Hi all. I have been digging around trying to find an answer to a few questions that are bugging me. I am hoping someone here can help.

1 – If you start an Async IO (BeginAccept) and the client hangs up what happens? The docs really don't tell you one way or the other. In my local testing many of the sockets are never getting released for some reason. I am not seeing my delegate called. I used an Interlock increment on each begin and a decrement on each delegate to track it and sometimes they come through, and sometimes they don't. Very odd. What is

## Re: Async Socket IO Question

the correct behavior?

2 – Why does the .Connected property not update? I have read the docs and tried lots of tests. It still does not appear to work.

```
try
{
// Attempt to ensure we are still connected...
byte[] temp = { 0x00 };
// I tried blocking and non-blocking didn't make a difference
sp.Client.Blocking = false;
// This is the MS recommended way. It ALWAYS returns 0 for me and
connected is not updated.
int res = sp.Client.Send(temp, 0, 0);
// I added this as an additional test and the same thing happens.
Always get back 0.
res = sp.Client.Receive(temp, 0, SocketFlags.None);
// If we get here we are still connected and alive...
}
catch( SocketException e )
{
if( e.NativeErrorCode.Equals(10035) )
{
// Still connected – the call would black
}
else
{
// We are disconnected
TimeoutOccured(ref sp);
return (false);
}
}
```

This app is a socket server that has remote clients connect. I can manually telnet to the app, watch it start a read, and then kill the telnet app. I know the socket is gone. I look in the process list and telnet is gone. But the reads and sends still report it is valid, and connected still reports true.

I do this test above in my routine prior to calling the beginreceive (I figure there is no use beginning a receive if the client hung up), and before sending data. Doesn't seem to make a difference.

What am I doing wrong?